

SYSTEM, METHOD, AND COMPUTER PROGRAM PRODUCT FOR USE OF LATTICES IN VALUING OPTIONS

Inventor: Elie Ayache

CROSS-REFERENCE TO RELATED APPLICATIONS

This application claims priority to International Patent Application No. PCT/IB00/01120, filed August 14, 2000, which claims priority to United States Provisional Patent Application Serial No. 60/148,318, filed August 11, 1999.

BACKGROUND OF THE INVENTION

Field of the Invention

[0001] The present invention relates generally to the valuation of financial instruments and more particularly to numerical methods for valuing stock options.

Background Art

[0002] Within the financial arts, an “option” is widely known as a contract that represents the right to buy or sell a specified amount of an underlying security at a specified price within a specified time. Once the contract is executed, the purchaser acquires a right, and the seller assumes an obligation. The underlying security (i.e., the subject of the option contract) can be, for example, stocks, bonds, futures, interests in real estate, commodities, etc. It is also not uncommon for the option to be underlain with more than one security (e.g., an option on a “basket” of different stocks).

[0003] An option is termed a “call” option if the contract gives the purchaser a right to buy the security at the specified price within the specified time. Alternatively, an option is termed a “put” option if the contract gives the purchaser a right to sell the security at the specified price within the specified

time. The option contract's specified price is termed the "strike price," while the option contract's specified time is termed the "maturity date."

[0004] In the case of a call stock option, the purchaser is guaranteed, at maturity, the ability to purchase the specified stock at the strike price. Thus, if the stock's market price at maturity were to be higher than the strike price, the seller (i.e., the party who sold the stock option) would have to pay the difference. If the buyer notices, at maturity, that the stock price in the open market is lower than the strike, the option will simply be allowed to expire unexercised. Alternatively, a put stock option guarantees the option holder the ability, at maturity, to sell the stock at the strike price, thus insuring against a future fall in the stock's price.

[0005] Within the world financial markets, it is important to mention one more distinction among options--there are American options and European options. An American option allows the holder to exercise the contract's rights anytime before the maturity date, whereas a European option can only be exercised at the maturity date. American-type options are those most popularly traded.

[0006] Given the above discussion, one can see how an option may serve as a form of insurance. The seller of the option (also referred to as the writer of the option) plays the role of the insurer and will ask for a premium to cover its liability. Thus, a problem arises as to exactly how to price an option.

[0007] The option pricing problem is that of determining a rational price for the option. That is, determining the premium that a rational buyer (i.e., not one hoping to deceive the insurance company) and a rational seller (i.e., not one ready to take inconsiderate amount of risk) are prepared to agree upon. The comparison of options pricing with insurance policy pricing suggests that the main concerns are estimating the probability of the stock's market price, at maturity, ending up higher or lower than the strike price, and by how much. The problem is further compounded by the fact that there may exist no consensus over such a probability among the parties, and that risk averse investors may value the option much more than those who are not risk averse.

[0008] A breakthrough in the problem of options pricing came from the development of a mathematical formula that showed how the rational price of an option could be made independent from any subjective elements--the estimated probability of the stock settling higher or lower, and the utility functions of the investors (i.e., their attitudes towards risk). That mathematical formula, as is now well-known in the relevant art(s), is the Black-Scholes equation. The Black-Scholes equation and model was first introduced in F. Black and M. Scholes, *The Pricing of Options and Corporate Liabilities*, Journal of Political Economy, 81 (May-June 1973), 637-59, which is incorporated herein by reference in its entirety.

[0009] The Black-Scholes formula for determining the price of a call option, C , using the five parameters essential to the pricing of an option: (1) the strike price K ; (2) the time to expiration t , (3) the underlying security price S ; (4) the volatility of the commodity σ ("sigma"); and (5) the prevailing risk-free interest rate r (the risk-free rate is the rate the safest bank deposit would earn or typically the rate of a U.S. government T-bill), is shown in EQUATION (1):

$$(1) \quad C = S * N(d_1) - Ke^{-(rt)} * N(d_2)$$

As will be apparent to one skilled in the relevant art(s), e is the exponential function--the inverse of the natural logarithm \ln --that is equal to, up to four significant decimal places, 2.7183. The variables d_1 and d_2 within EQUATION (1) are expressed as shown in EQUATIONS (2A) and (2B), respectively:

$$(2A) \quad d_1 = \frac{\ln\left(\frac{S}{K}\right) + \left(r + \frac{\sigma^2}{2}\right)t}{\sigma\sqrt{t}}$$

$$(2B) \quad d_2 = d_1 - \sigma\sqrt{t}$$

The function " $N(\)$ " is the cumulative standard normal distribution function, which is well known in the relevant art(s).

[0010] Having presented the Black-Scholes formula for a call option, EQUATION (3) describes the expression for the price P of a put option:

$$(3) P = C - S + Ke^{-rt}$$

[0011] Inherent in the Black-Scholes formulation is the assumption that the writer of the option (e.g., the call seller) will hedge his liability by buying a certain amount of the stock(s) underlying the option prior to any price rally. This assures the writer will be able to simply hand over the stock(s) to the buyer at the maturity of the contract. The difficulty of course is that the option seller cannot foretell the movement of the stock during the life of the option contract. The seller will naturally like to avoid holding stock(s) that no other party--least of all the option buyer--would desire to purchase should the share prices plummet at maturity. Thus, the writer must constantly monitor the evolution of the shares, gradually getting rid of the initial hedge if prices are seen to go down, or gradually building up the hedge if prices are seen to go up. This scheme is often termed a "replication strategy" or "dynamic hedge program." Therefore, the option seller should account for the replication strategy when setting the price of the option (i.e., ask for an amount of money that the strategy is expected to cost).

[0012] As suggested above, of the many factors that impact the price of a stock option, the most relevant is the degree of agitation of the share along its trajectory, which is called its "volatility," σ . Trajectory, as used herein, refers to a plot of a stock's market price over time. That is, the degree to which the stock's price goes up or down over the course of its trading is most relevant rather than the actual up or down direction of the trajectory, or equivalently, the prior chances that a trajectory has to go up or down. In other words, the actual direction has no impact on the option price. Mathematically this is expressed as follows: Let S be the stock price and dS its variation over an infinitesimal period of time. The infinitesimal percentage return of the stock will then be dS/S .

[0013] One cannot tell with certainty how the return of the stock is going to behave over time. One can only tell where one expects the return to be over the next infinitesimal period of time, it being understood that a certain noise, or

indeterministic factor, is bound to spoil one's expectation. In other words, the temporal evolution of the return will include two parts: a deterministic part, called the "drift" that represents one's expectation about the future infinitesimal movement, and an indeterministic part that represents the amount of error that one will be found to have committed in reality. The return can then be expressed using EQUATION (4):

$$(4) \quad \frac{dS}{S} = \mu dt + \sigma dZ$$

where μdt is the expected return over the infinitesimal period of time, and σdz is the infinitesimal error. Return can be thought of as a moving object (e.g., a rocket), whose instantaneous projected speed is μ , and whose fluctuation around its projected trajectory is accounted for by σdz .

[0014] The main issue in theoretically modeling such a trajectory will be one of modeling the error term. As is usually the case when the sources of error and uncertainty are many, and they are many in the case of stocks and financial markets in general, the assumption one makes is that those errors are normally distributed. In other words, the assumption behind the Black-Scholes model is that return will be distributed, over the next infinitesimal period of time, according to a "bell curve," with mean μdt and standard deviation $\sigma \sqrt{dt}$ as shown in FIG. 1. The evolution of the return of the stock, when modeled in this fashion is termed Brownian motion (i.e., a phenomenon of constant erratic motion which may be mathematically modeled) by those skilled in the relevant art(s).

[0015] The greater the volatility parameter, σ , the greater the dispersion of the return around its expected value, the greater the jaggedness of the corresponding overall trajectory, and thus, the greater the difficulty to track the share price. Also, the instantaneous expected return (or drift) parameter, μ , governs the overall trend of the trajectory. Thus, the greater the drift, the higher prices should be expected to go. Conversely, the smaller the drift (it could of course turn negative), the lower prices should be expected to go.

[0016] When investors expect a share to perform well, (i.e., when they assign a greater probability to its trajectory going up rather than down), it is the drift parameter, μ , that they are predicting. The Black-Scholes formulation has shown is that μ is irrelevant to the option pricing problem. Only the volatility, σ , which measures the more or less orderly way in which the share will follow the investors' expectations, will affect the option price. The mathematical reasoning for this is given below.

[0017] Because the state variables are the stock price S and time t , one would expect the option price C to be function of S and t as shown above in EQUATION (1). This relationship is given more simply by the expression $C(S, t)$.

[0018] When a function has a stochastic variable following the above Brownian motion as one of its state variables, Ito's rule for stochastic differentiation states that the differential of C is given by EQUATION (5):

$$(5) \quad dC = \left(\frac{\partial C}{\partial t} + \mu S \frac{\partial C}{\partial S} + \frac{1}{2} \sigma^2 S^2 \frac{\partial^2 C}{\partial S^2} \right) dt + \sigma S \frac{\partial C}{\partial S} dZ$$

The Black-Scholes hedge assumption then amounts to noting that if a certain amount, Δ , of the share were held with the option as a hedge against adverse movements of the share, the resulting portfolio:

$$\Pi = C + \Delta S$$

will yield the following return over the infinitesimal period of time shown in EQUATION (6):

$$(6) \quad d\Pi = dC + \Delta dS$$

[0019] Hence, EQUATION (7), owing to the expression of dS given by the Brownian motion:

$$(7) \quad d\Pi = \left(\frac{\partial C}{\partial t} + \mu S \frac{\partial C}{\partial S} + \frac{1}{2} \sigma^2 S^2 \frac{\partial^2 C}{\partial S^2} + \Delta S \mu \right) dt + \left(\sigma S \frac{\partial C}{\partial S} + \Delta \sigma S \right) dZ$$

[0020] The Black-Scholes formulation then notes that if Δ were chosen equal to

$-\frac{\partial C}{\partial S}$, the indeterministic component of $d\Pi$ would vanish, and EQUATION (7)

would simply be expressed as EQUATION (8) (the term involving μ , has also dropped off):

$$(8) \quad d\Pi = \left(\frac{\partial C}{\partial t} + \frac{1}{2} \sigma^2 S^2 \frac{\partial^2 C}{\partial S^2} \right) dt$$

No indeterministic noise would consequently threaten the growth of portfolio Π . In other words, Π would become a riskless portfolio. A simple arbitrage argument then establishes that Π would then have to grow at the riskless rate of interest r (the interest rate that the safest bank deposit would earn). Otherwise, supposing that Π grows at a rate greater than r one could borrow money at the rate r , invest it in the riskless portfolio Π , and earn the riskless difference. Else, if Π grows at a rate smaller than r , one could short sell the portfolio Π , deposit the proceeds in the bank account earning r , and also make a riskless profit. Consequently the following must be true:

$$d\Pi = r\Pi dt$$

hence EQUATION (9):

$$(9) \quad \frac{\partial C}{\partial t} + \frac{1}{2} \sigma^2 S^2 \frac{\partial^2 C}{\partial S^2} = rC + r\Delta S = rC - r \frac{\partial C}{\partial S} S$$

[0021] Finally, $C(S, t)$ must verify the following differential EQUATION (10):

$$(10) \quad \frac{\partial C}{\partial t} + r \frac{\partial C}{\partial S} S + \frac{1}{2} \sigma^2 S^2 \frac{\partial^2 C}{\partial S^2} = rC$$

[0022] The Black and Scholes option pricing formula is an analytical solution of the partial differential EQUATION (10). S and t are the state variables (i.e., it is with respect to them that the partial differentiation is taken), and σ and r are the

sole parameters. No trace is left of the expected return μ and its underlying probability or utility estimations.

[0023] The option pricing problem, however, admits of a probabilistic interpretation. Because the Black and Scholes argument has established that the option price does not depend on the investors' subjective preferences, estimations of probability, and/or attitudes towards risk, the option price will, in particular, be the price that would have obtained had such preferences and attitudes to risk never existed in the first place. In other words, if all the investors were neutral to risk (a hypothetical world which is labeled the "risk-neutral world" in the relevant art(s)), the option price would nevertheless be the same. The only difference between such a risk neutral world and the actual world, would be that investors would not expect the shares to grow at a rate higher than their risk-free rate investments (e.g., a safe bank deposit). Because these investors are indifferent to risk, they would not ask for a premium for holding a risky share. Hence the evolution of the return of the shares in the risk-neutral world will be simply given by the following stochastic process of EQUATION (11):

$$(11) \quad \frac{dS}{S} = rdt + \sigma dZ$$

[0024] In view of such a process, one would only need to estimate the value of σ to be able to draw realizations of the Brownian paths such as the ones drawn above, for the interest rate r is known. Every such realization will produce a different "hit" S_T (i.e., a different final stock price) at the maturity of the option. Remembering that the call option pays at maturity the difference between S_T and the strike price, K if S_T is greater than K , else zero, and noting that $(S_T - K)^+$ the payoff (i.e., $(S_T - K)^+ = \max(S_T - K, 0)$), one may compute the expected value of the option payoff using EQUATION (12):

$$(12) \quad E(\text{payoff}) = \sum_{\text{all paths}} \text{prob}_{\text{path}} (S_T - K)^+$$

where the summation is taken over all the paths that the Brownian motion of EQUATION (11) can generate.

[0025] Stochastic calculus establishes that the solution of the partial differential EQUATION (10) above (i.e., the price of the option), can be represented as the discounted expected value of the option payoff under the Brownian motion of EQUATION (13):

$$(13) \quad C = e^{-rT} \sum_{\text{all paths}} \text{prob}_{\text{path}} (S_T - K)^+$$

Thus, there are two different methods of computing the price of an option, either solve the partial differential EQUATION (10), or compute expected values under all possible paths using EQUATION (13).

[0026] Difficulties, however, prevent the option pricing problem from being as simple as it first seems. The Black-Scholes analytical solution is valid only when the interest rate r and the volatility parameter σ are constant. One may be interested in pricing options under a more general Brownian motion such as EQUATION (14):

$$(14) \quad \frac{dS}{S} = r(S,t)dt + \sigma(S,t)dZ$$

where the parameters now themselves depend on time and the stock price. A great deal of the literature in the relevant art(s) that followed Black and Scholes' 1973 paper has studied the option pricing problem under these more general assumptions. No analytical solutions, however, are generally available when the stochastic process is given by EQUATION (14), so one has to resort to a numerical resolution of the partial differential EQUATION (10). Alternatively, one can attempt to compute expected values. The only difficulty in that case is that the number of paths generated by a Brownian motion is infinite. Thus, one cannot practically take summations over all paths. One may choose to select a finite number of representative paths to perform the summation--a process known as "discretization" of the stochastic process.

[0027] Thus, two different schools of option researchers exist. Those who would rather numerically solve the partial differential EQUATION (10), and those who would rather seek a discretization of the driving stochastic process--known as the "tree" or "lattice" approach. The tree or lattice approach, however, is better suited to the option pricing problem because it can accommodate not only European options easily (like the Black-Scholes model), but can easily handle American options as well.

[0028] Therefore, what is needed is a system, method, and computer program product that solves the option pricing problem while accounting for drift and volatility parameters of the most general form (i.e., EQUATION (14)). The system, method, and computer program product should use the lattice approach and handle the case where the number of dimensions is greater than one (i.e., when there is more than one stochastic process underlying the option). The system, method, and computer program product should also seek the most economical and efficient lattice discretization and thus, achieve the maximum economy in terms of computing time and resources.

BRIEF SUMMARY OF THE INVENTION

[0029] The present invention is a system, method, and computer program product for use of lattices in valuing options. More specifically, the method prices the current value of an option consisting of a plurality of underlying assets (i.e., a basket option). The method includes the steps of receiving inputs indicative of the spot value of the underlying assets and the diffusion parameters. The diffusion parameters include the correlation, ρ , of each asset to each other asset within the basket, and the volatility, σ , of each of the plurality of assets. Then, inputs indicative of the risk free interest rate, r ; the drift function, $\mu(X, t)$, and the probability parameter, p are received.

[0030] The method further includes receiving inputs indicative of the desired time slices and the maturity date of the option. Then a lattice is constructed using

the constant parameters, wherein the lattice is comprised of an elementary lattice cell structure for each of the time slices. Once the lattice is constructed, it is traversed, in a backwards fashion, in order to calculate the value of the option from said date of maturity to the present.

[0031] One advantage of the present invention is that it can calculate the price of any contingent claim that is underlain by two or three driving Brownian motions (i.e., the pricing of spread and basket options), the pricing of options and convertible bonds under stochastic interest rates, the pricing of interest rate derivatives when two or three stochastic factors govern the evolution of the yield curve, etc.

[0032] Another advantage of the present invention is that it can be used as a tool in managing a portfolio of multi-factor options, and assessing (at all times and in all market conditions) the value of the options. The present invention can also assists in the hedging and re-hedging (at all times) of a portfolio. That is, buying and selling orders of the underlying shares or other liquid market instruments can be generated in order to achieve neutrality with respect to adverse market movements.

[0033] Another advantage of the present invention is that it can assists in the assessment of risks of options held in a portfolio by providing an evaluation of hypothetical losses (or gains) that the portfolio incurs on sudden (or gradual) market and volatility shifts.

[0034] Yet another advantage of the present invention is that, more generally, it serves any purpose that would require the discretization of two or three-dimensional Brownian motions with general drift and volatility parameters (e.g., physicists interested in the random motion of a particle in 3D space can utilize the lattices of the present invention as an efficient illustration of that motion, etc.).

[0035] Further features and advantages of the invention as well as the structure and operation of various embodiments of the present invention are described in detail below with reference to the accompanying drawings.

BRIEF DESCRIPTION OF THE FIGURES

[0036] The features and advantages of the present invention will become more apparent from the detailed description set forth below when taken in conjunction with the drawings in which like reference numbers indicate identical or functionally similar elements. Additionally, the left-most digit of a reference number identifies the drawing in which the reference number first appears.

[0037] Fig. 1 is a diagram illustrating a standard bell curve;

[0038] Fig. 2 is a diagram illustrating a Cox tree showing the constant size paths representing up and down movements of a stock price;

[0039] Fig. 3 is a binomial lattice showing the constant size paths representing up and down movements of a stock price;

[0040] Fig. 4 is a binomial lattice showing the evolution of the stock price over a single time interval;

[0041] Fig. 5 is a binomial lattice showing the corresponding binomial fork of a stock price is the discounted expected value of its two terminal values;

[0042] Fig. 6 is a tree illustrating the call price of a stock option payoff at maturity;

[0043] Fig. 7 is a trinomial tree illustrating variable drift and volatility parameters;

[0044] Fig. 8 is a trinomial Hull and White lattice illustrating the permutations for a basket option containing two assets;

[0045] Fig. 9 is a flowchart representing the lattice creation process according to an embodiment of the present invention;

[0046] Fig. 10 is a flowchart representing the lattice traversal process according to an embodiment of the present invention;

[0047] Fig. 11 is a diagram of an integer hexagon produced according to an embodiment of the present invention;

[0048] Fig. 12 is a diagram of an physical hexagon produced according to an embodiment of the present invention;

[0049] Fig. 13 is a diagram of a rhombic dodecahedral lattice produced according to an embodiment of the present invention;

[0050] Fig. 14 is a block diagram of an example computer system useful for implementing the present invention;

[0051] Fig. 15 is a rhombic dodecahedron produced according to an embodiment of the present invention; and

[0052] Fig. 16 is a the nucleus of the rhombic dodecahedron (Lord Kelvin's solid) produced according to an embodiment of the present invention.

DETAILED DESCRIPTION OF THE INVENTION

Table of Contents

- I. Overview
- II. The Lattice Approach
 - A. The Binomial Lattice
 - B. The Trinomial Lattice
- III. The Model: PI's Extension of the Lattice Approach
- IV. Example of System Operation - Lattice Creation
- V. Example of System Operation - Lattice Traversal
- VI. Details of Lattice Geometry - Two Dimensional Case
 - A. Inputs
 - B. Determination of the Geometry of the Elementary Cell
 - C. Building of the Lattice
 - D. Traversing the Lattice
- VII. Details of Lattice Geometry - Three Dimensional Case
 - A. Inputs
 - B. Determination of the Geometry of the Elementary Cell
 - C. Building of the Lattice
 - D. Traversing the Lattice
- VIII. Computational Efficiency of the Present Invention
- IX. Example Implementations
- X. Conclusion

I. Overview

[0053] The present invention is directed to a system, method, and computer program product for use of lattices in valuing options. For example, the method allows the pricing of options where more than one asset is involved (i.e., a basket option). In an embodiment, the present invention is provided as a tool to users (either individuals or members of a trading firm) interested in valuing such options.

[0054] In an alternate embodiment of the present invention, a trading organization may provide a brokerage desk that facilitates option trades for clients as well as providing an interactive World-Wide Web site accessible via the global Internet for pricing model and trade execution services. Such an infrastructure may be an organized exchange for options.

[0055] The present invention is described in terms of the above example. This is for convenience only and is not intended to limit the application of the present invention. In fact, after reading the following description, it will be apparent to one skilled in the relevant art(s) how to implement the following invention in alternative embodiments (e.g., valuing other financial instruments or, more generally, illustrating any Brownian motion).

II. The Tree and Lattice Approach

A. The Binomial Lattice

[0056] The first, simplest lattice was devised by Cox, Ross and Rubinstein as described in J.C. Cox *et al.*, *Option Pricing: A Simplified Approach*, Journal of Financial Economics, 7, (October 1979), 229-63, which is incorporated herein by reference in its entirety. It assumes that the drift parameter, μ , and volatility parameter, σ , are constant, and therefore only improves the Black-Scholes formulation with respect to the pricing of American options.

[0057] The Cox tree further assumes that the stock is allowed to move in *finite* intervals of time, and selects among all possible paths only those that can be represented as an arbitrary concatenation of up and down movements of constant size as shown in FIG. 2. If the life span of the option is sliced into n equal time intervals, there will be a total of 2^n such possible “discrete” paths.

[0058] Supposing that the probability of an elementary up movement is p and that of an elementary down movement is $(1 - p)$, the probability of any one path is given by EQUATION (15):

$$(15) \quad \pi(path) = \Pi p^k (1 - p)^{n-k}$$

where k is the number of up jumps taking place over the whole path. This suggests that the option price, computed as the discounted expected value of its payoff, will have to be a sum of 2^n terms as shown in EQUATION (16):

$$(16) \quad C = e^{-rT} \sum \Pi p^k (1 - p)^{n-k} (S_T - K)^+$$

[0059] This calculation, however, turns out to be much simpler. If one were to draw the grid underlying all possible paths, one would obtain a lattice, called a ‘binomial lattice’ as shown in FIG. 3. Owing to the regularity and symmetry of the lattice, the final possible values of the stock will be known beforehand, and so would the possible option payoffs.

[0060] Computing the option price will then proceed recursively in the following fashion. Calling u and d , respectively, the sizes of the elementary up and down movements, the evolution of the stock price over a single time interval can be pictured as shown in FIG. 4.

[0061] Requiring that the binomial lattice be an approximation of the Brownian motion with drift parameter μ and volatility parameter σ can be shown to impose EQUATIONS (17) and (18):

$$(17) \quad pu + (1 - p)d = e^{rdt}$$

$$(18) \quad pu^2 + (1 - p)d^2 = \sigma^2 dt + e^{2rdt}$$

where dt is the size of the finite time interval. EQUATIONS (17) and (18) are called, respectively, the first moment and second moment equations. Assuming that $d = 1/u$, this yields the approximate solution to EQUATIONS (17) and (18):

$$\begin{aligned} u &= e^{\sigma\sqrt{dt}} \\ d &= e^{-\sigma\sqrt{dt}} \\ p &= \frac{e^{rdt} - d}{u - d} \end{aligned}$$

Because the option price C in the corresponding binomial fork is the discounted expected value of its two terminal values as shown in FIG. 5, this yields EQUATION (19):

$$(19) \quad C = e^{-rdt} [pC_u + (1-p)C_d]$$

In order to compute the call price in the full-blown lattice, one has to express the option payoff at maturity as shown in FIG. 6 and then roll back towards the origin of the lattice, computing the option value at each node as the discounted expected value of the corresponding binomial fork. As the total number of nodes of a binomial lattice is:

$$\frac{(n+1)(n+2)}{2}$$

and the value at each node is only the sum of two terms, the summation is far less than the dreaded summation of 2^n terms as in EQUATION (16).

[0062] Despite its simplicity, the binomial lattice has drawbacks. The expressions of u , d , and p above are only approximate. The exact values are functions of both the drift and the volatility. So one could not accurately discretize a Brownian motion with varying drift and volatility coefficients without altering the values u , and p throughout the lattice. This would alter its simple structure.

B. The Trinomial Lattice

[0063] One alternative, when the drift and volatility parameters are required to be variable, is the trinomial lattice. The trinomial lattice, as its name suggest, has three branches emanating from each node. Calling u , d , and m , respectively, the sizes of the elementary up, down, and middle movements, the evolution of the stock price over a single time interval can be shown in FIG. 7 (compare to FIG. 4). Thus, the transition probabilities are now p_u , p_d , and p_m , where $p_m = 1 - p_u - p_d$.

[0064] This provides an extra degree of freedom which allows u and d to remain constant throughout the lattice, while the probabilities solve the first and second moment equations and account for the varying character of the drift and volatility parameters:

$$u = e^{\sigma\sqrt{3dt}}$$

$$d = \frac{1}{u}$$

$$p_u = \sqrt{\frac{dt}{12\sigma^2}} \left(r - \frac{\sigma^2}{2} \right) + \frac{1}{6}$$

$$p_m = \frac{2}{3}$$

$$p_d = -\sqrt{\frac{dt}{12\sigma^2}} \left(r - \frac{\sigma^2}{2} \right) + \frac{1}{6}$$

(See Hull, John, *Options, Futures, and Other Derivative Securities*, Third Edition, Prentice Hall (Englewood Cliffs, N.J), 1997, ISBN 0138874980, p. 360, which is incorporated herein by reference in its entirety.)

[0065] In Hull and White, *Branching Out*, RISK, July 1994, which is incorporated herein by reference in its entirety, the use of trinomial lattices was first promoted. Moreover, they were the first to solve the problem of negative probabilities (p_d can turn negative if μ is sufficiently large) by the alternative branching technique which the present invention addresses.

[0066] Also, when more than one stochastic variable is underlying the option, Hull and White simply suggested to take the tensor product lattice, which is not the most efficient. For example, consider an option on a basket of two stocks S and W. Hull and White recommends that for each node of the lattice, nine nodes be generated to represent the permutations of each stock going up, down, or remaining the same (i.e., middle). Thus, the Hull and White lattice, in this instance, requires nine probabilities to be calculated at each node. This is illustrated in FIG. 8.

III. *The Model: Present Invention's Extension of the Lattice Approach*

[0067] The present invention extends Hull and White's trinomial technique to obtain the most economical and efficient lattice discretization for higher dimensional Brownian motions of the most general form (i.e., EQUATION (14)). Consider TABLE 1, below, which highlights the present invention's efficient computation:

# OF UNDERLYING ASSETS IN OPTION BASKET	HULL & WHITE NODES	PRESENT INVENTION NODES	PRESENT INVENTION LATTICE ELEMENTARY CELL SHAPE
2	9	7	Hexagon
3	27	15	Rhomba- Dodecahedron
⋮			
N	3^N	$(2^{N+1}) - 1$	$(2^{N+1}) - 2$

TABLE 1

Thus, the present invention will produce a hexagonal structure (six nodes plus the middle for a total of seven nodes) in computing the option price of a basket containing two assets. Accordingly, the present invention will produce a rhomba-dodecahedron (fourteen nodes plus the middle for a total of fifteen nodes) in computing the option price of a basket containing three assets. The present

invention aims to use a simpler tree than 3^n and assures the probabilities are positive no matter the value of the drift, μ .

IV. Example of System Operation - Lattice Creation

[0068] Referring to FIG. 9, a flowchart representing the lattice creation process 900 of the present invention is shown. More specifically, lattice creation process 900 illustrates the case where the present invention is used to price an option on a basket containing multiple assets (e.g. stocks S_1 and S_2 --the two asset row of TABLE 1). Lattice creation process 900 begins at step 902 with control passing immediately to step 904.

[0069] In step 904, lattice creation process 900 receives inputs representing the spot value of the assets S_1 and S_2 . In step 906, the maturity date, T , of the basket option is entered. In an embodiment of the present invention, the maturity date is specified in terms (or a fraction) of a year (i.e., for a six month maturity date, $T = 0.5$). In step 908, the prevailing risk-free interest rate, r , is entered. In step 910, the respective volatility of the underlying assets are entered (e.g., σ_1 and σ_2 for S_1 and S_2 , respectively). Also, the correlation, ρ , between each two pairs of assets are entered. These three inputs are collectively known as the "diffusion parameters."

[0070] In step 912, the number of time slices, N , the life span of the option is sliced into equal time intervals is entered. As will be apparent to one skilled in the relevant art(s), the greater the value of N , the more accurate the discretization (i.e., greater the accuracy of the present invention), but with the associated increase in computation time and resources. Moreover, as N approaches infinity, the accuracy of the output price of the present invention approaches the accuracy of the Black-Scholes model. In step 914, the value of elementary time step, Δt is calculated, which is T/N .

[0071] In steps 916, the drift function, $\mu(X, t)$, is entered, and in step 918, the probability parameter, p , is entered.

[0072] In step 920, the lattice specified in TABLE 1 is generated using a recursive procedure going forward. For example, in the case of an option basket of two underlying assets, the lattice generated in step 920 would be an initial node (the “origin” or “root”) which generates an initial hexagon. Each node of the initial hexagon would then recursively spawn another hexagon N times (i.e., the number of time slices entered). Each of the seven nodes generated by the probability distribution function (e.g., see EQUATION (31) as explained in detail below). Lattice creation process then ends as indicated by step 922.

V. Example of System Operation - Lattice Traversal

[0073] Referring to FIG. 10, a flowchart representing the lattice traversal process 1000 of the present invention is shown. More specifically, lattice traversal process 1000 illustrates the case where the present invention is used to price an option on a basket containing multiple assets (e.g. stocks S_1 and S_2). That is, process 1000 traverses the lattice created by process 900 explained above with reference to FIG. 9. Lattice traversal process 1000 begins at step 1002 with control passing immediately to step 1004.

[0074] Once the lattice is created as shown in FIG. 10, there exist a record, for each time slice, of the nodes which constitute that time slice. Thus, in step 1004, the value at maturity of the option (time slice $t = N$) is calculated. Next, the value of the option at every time slice ($t = N-1, -2, \dots, 0$) can be calculated using the following steps 1008-1026 as indicated by step 1006. In step 1008, the spacial coordinates for each of the nodes of a time slice’s lattice structure (e.g., the seven nodes of the hexagon for a basket option of two assets) is obtained. In step 1010, the value of the assets (e.g., S_1 and S_2) are calculated at each of the nodes. In step 1012, the drift vector, μ , is calculated at each node using the drift function inputted in step 916 of lattice creation process 900.

[0075] In step 1014, the coordinates of the children of each node is obtained. In step 1016, the stored value of the option at each of the children nodes is read. In

step 1018, the probabilities for each node are calculated. In step 1020, the option price, C , is calculated for the time slice. In step 1022, the price is stored. This process repeats until no more time slices are left to traverse as indicated by step 1006.

[0076] Once all time slices have been traversed and the root node is reached, in step 1024, (time slice $t = 0$), the price, C , for the original node is calculated. In step 1026, the current price of the option is outputted and process 1000 ends as indicated by step 1028.

[0077] As will be apparent to one skilled in the relevant art(s), the present invention provides a computational advantage in that when the value at time slice $t = n-2$ is calculated, the values for time slice $t = n$ may be discarded as they are no longer needed. Thus, when the values for $t = n-4$ are calculated, the values for $t = n-2$ are no longer needed, and so forth.

VI. Details of Lattice Geometry - Two Dimensional Case

[0078] Having generally described the lattice creation process 900 and lattice traversal process 1000 above, a more detailed description is now given for the case where pricing a basket option which includes two underlying assets is desired. Further, example C++ programming language source code that implements the lattice creation process 900 and lattice traversal process 1000, as described in detail below for the two-dimensional case, is shown in APPENDIX A.

A. Inputs

[0079] The first consideration is the two underlying stochastic variables representing each asset in the basket, x_1 and x_2 (e.g., stocks S_1 and S_2 presented above with reference to FIGS. 9 and 10). These two variables form one underlying stochastic vector:

$$X = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$$

The stochastic processes in the risk-neutral world are:

$$dx_1 = \mu_1(X, t)dt + \sigma_1 dz_1$$

$$dx_2 = \mu_2(X, t)dt + \sigma_2 dz_2$$

with correlation ρ between dz_1 and dz_2 . The volatility parameters, σ_1 and σ_2 , and the correlation coefficient, ρ , are constant, but the drift vector:

$$\mu = \begin{pmatrix} \mu_1 \\ \mu_2 \end{pmatrix}$$

depends on both the state X and time t . The initial state at time $t = 0$ is:

$$X_0 = \begin{pmatrix} x_1^0 \\ x_2^0 \end{pmatrix}$$

[0080] The present invention aims to price, at time $t = 0$, a contingent claim C (i.e. a derivative instrument whose payoff is contingent upon the underlying vector X), of time to maturity T and payoff at maturity $P(X)$. The price at time $t=0$, is denoted by $C(X_0, 0)$. If the derivative instrument is an American-type option, its time t , state X , price $C(X, t)$ must verify:

$$C(X, t) \geq I(X)$$

This is because the function “ $I(\cdot)$ ” is the lower bound of the American-type option. Typically, in the simplest case, this is the intrinsic value of the option (i.e., $(S - K)$, or for convertible bonds, $(R * S)$, where R is the conversion ratio). Optimality in exercising options requires that one check, while traversing the lattice, to ensure that the stored value of the option is always greater than the intrinsic value. If this were not the case, the holder would simply exercise the

option, and retain the greater (intrinsic) exercise value.. Thus, as described below, the greater of the two values is always stored while traversing the lattice. Hence, exercising the option is optimal in those nodes where the computed value, C , is replaced by the lower bound of the American-type options.

[0081] As shown in FIG. 9, inputs of the lattice building process 900 are: (a) the diffusion parameters, ρ, σ_1, σ_2 ; (b) the complete drift function:

$$\mu(X, t) = \begin{pmatrix} \mu_1(X, t) \\ \mu_2(X, t) \end{pmatrix}$$

(c) the initial value of the underlying state vector, X_0 ; (d) the value of the short interest rate r (or the interest rate term structure - yield curve or forward curve); (e) the number of time steps, N , one wishes to take; and (f) the probability parameter p .

[0082] As will be appreciated by one skilled in the relevant art(s), the short rate, r , will not be an input if it is itself one of the underlying stochastic variables.

B. Determination of the Geometry of the Elementary Cell

[0083] In the two-dimensional (i.e., two stochastic variables) case, the elementary cell is a centered hexagon (see TABLE 1 above). However, it must be geometrically deformed in order to reflect the diffusion matrix:

$$\Sigma = \begin{pmatrix} \sigma_1^2 & \rho\sigma_1\sigma_2 \\ \rho\sigma_1\sigma_2 & \sigma_2^2 \end{pmatrix} \Delta t$$

The transformation matrix A is:

$$A = \sqrt{\frac{2}{P}} P \sqrt{\Lambda}$$

where:

$$\Sigma = P\Lambda P^*$$

(The symbol “*” denotes the transpose of a matrix.) That is, the matrix:

$$\Lambda = \begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{pmatrix}$$

is the matrix of eigenvalues of Σ , and P the matrix of eigenvectors, which are computed by standard techniques. The present invention's lattice (i.e., the set of states or nodes:

$$X = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$$

retained for the discretization) is a deformed hexagonal lattice. However, a straight computational lattice, E, is defined where coordinates of the nodes have integer values and correspond one to one to the real "physical" nodes:

$$E = \begin{pmatrix} i_1 \\ i_2 \end{pmatrix} \leftrightarrow X = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$$

through the transformation:

$$\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = A \begin{pmatrix} 1 & 1/2 \\ 0 & \sqrt{3}/2 \end{pmatrix} \begin{pmatrix} i_1 \\ i_2 \end{pmatrix}$$

[0084] In the computational plane, the hexagon is as shown in FIG. 11. The hexagon of FIG. 11 has vertices having the following integer co-ordinates:

$$\begin{pmatrix} 1 & 0 & -1 & -1 & 0 & 1 \\ 0 & 1 & 1 & 0 & -1 & -1 \end{pmatrix}$$

The "integer hexagon" of FIG. 11 is then transformed into the typical "physical hexagon" shown in FIG. 12 using the following matrix:

$$H = \begin{pmatrix} 1 & 1/2 \\ 0 & \sqrt{3}/2 \end{pmatrix}$$

The “physical hexagon” shown in FIG. 12 is then further transformed by matrix A to reflect the diffusion parameters.

C. Building of the Lattice

[0085] The building of the lattice (i.e., step 920 of process 900) is a recursive procedure running forward in time. It starts with the origin (or root):

$$X_0 = \begin{pmatrix} x_1^0 \\ x_2^0 \end{pmatrix}$$

and progressively “marks,” for each time slice n ($0 \leq n \leq N$), the nodes of the computational lattice E which will belong to this time slice. It is the drift function that governs this process. The recursive mechanism can be described by the following five-step process. If nodes of time slice $n - 1$ are given by the following set:

$$(20) \quad E_{n-1} = \{ E_{n-1}^k \}_{1 \leq k \leq K_{n-1}}$$

where K_{n-1} is the number of nodes, the following steps are performed:

[0086] First, for each node, E_{n-1}^k , its physical correspondent X_{n-1}^k is found by utilizing EQUATION (21):

$$(21) \quad X_{n-1}^k = X_0 + AHE_{n-1}^k$$

(This means the root X_0 has the computational correspondent $E_0 = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$)

[0087] Second, the drift vector at X_{n-1}^k , is computed using EQUATION (22):

$$(22) \quad \mu(X_{n-1}^k, (n-1)\Delta t) = \begin{pmatrix} \mu_1(X_{n-1}^k, (n-1)\Delta t) \\ \mu_2(X_{n-1}^k, (n-1)\Delta t) \end{pmatrix}$$

[0088] Third, the drift vector directs, by use of EQUATION (23), to a point Z_{n-1}^k somewhere in the physical plane:

$$(23) \quad Z_{n-1}^k = X_{n-1}^k + \mu_{n-1}^k$$

[0089] Fourth, the node of the lattice that is "closest" to Z_{n-1}^k must then be determined. This closest node is termed the "target node", Y_{n-1}^k . The target node is the candidate node for occupying the center of the cell of descendants of X_{n-1}^k . This means once the target node is determined, the other six children nodes of X_{n-1}^k will be determined as well, as vertices of the hexagon surrounding Y_{n-1}^k .

[0090] The determining criterion is the point where the drift points to, Z_{n-1}^k , and lies in the nucleus of the target cell (i.e. Y_{n-1}^k must be the center of the hexagon in whose nucleus Z_{n-1}^k lies). Calling $\begin{pmatrix} z_1 \\ z_2 \end{pmatrix}$ the physical coordinates of

Z_{n-1}^k , its coordinates in the computational plane are given by EQUATION (24):

$$(24) \quad e = \begin{pmatrix} e_1 \\ e_2 \end{pmatrix} = H^{-1} A^{-1} \begin{pmatrix} z_1 - x_1^0 \\ z_2 - x_2^0 \end{pmatrix}$$

These will not generally be integer values. However, the four nodes of the computational lattice (i.e., with integer coordinates) which will surround

$\begin{pmatrix} e_1 \\ e_2 \end{pmatrix}$ are desired. These four nodes are given by EQUATION (25):

$$(25) \quad (f_1 \ f_2 \ f_3 \ f_4) = \begin{pmatrix} \text{int}(e_1) & \text{int}(e_1)+1 & \text{int}(e_1) & \text{int}(e_1)+1 \\ \text{int}(e_2) & \text{int}(e_2) & \text{int}(e_2)+1 & \text{int}(e_2)+1 \end{pmatrix}$$

where $\text{int}(e_1)$ is the integer value of e_1 .

[0091] The target node, Y_{n-1}^k , will be the one among these four nodes that is closest to Z_{n-1}^k . As used herein, "closest" means nearest in terms of Euclidian distance as measured in the intermediary plane where the hexagonal cell is the traditional straight hexagon as depicted in FIG. 12. This is not the physical plane, nor is it the computational plane. Rather, this distance is measured in the plane which is the image by H of the computational plane E . In other words, the f_i among the four candidates above such that the Euclidian distance given by EQUATION (26) is minimal is chosen:

$$(26) \quad \|H(e) - H(f_i)\|$$

[0092] The selected f_i will then represent the coordinates in the computational plane of the sought after target node Y_{n-1}^k . As for the six other children, their coordinates in the computational plane will be given by EQUATION (27):

$$(27) \quad f_i + \begin{pmatrix} 1 & 0 & -1 & -1 & 0 & 1 \\ 0 & 1 & 1 & 0 & -1 & -1 \end{pmatrix}$$

[0093] Fifth, the above procedure is repeated for each one of E_{n-1}^k , and discarding repetitions, will produce the list of the nodes of the computational lattice which will constitute time slice n :

$$E_n = \{E_n^k\}_{1 \leq k \leq K_n} = \bigcup_{1 \leq k \leq K_{n-1}} \{\text{seven children of } E_{n-1}^k\}$$

D. Traversing the Lattice

[0094] Once the building the lattice is completed, a record for each time slice of the nodes constituting it will exist. More specifically, the complete list of nodes of the last time slice N is available:

$$E_N = \{E_N^k\}_{1 \leq k \leq K_N}$$

First, the payoff of the derivative instrument is then computed at this last time slice $t = N$. For each one of the nodes, its physical coordinates are determined by using the EQUATION (21):

$$X_N^k = X_0 + AHE_N^k$$

and the corresponding payoff is computed by $P(X_N^k)$. This is the value of the derivative instrument which will be stored in time slice $t = N$ at node E_N^k . Rolling back in the tree will then inductively take place as follows.

[0095] Assuming values of the derivative instrument have been computed and stored at all nodes E_n^k of time slice n , step back to time slice $n-1$. Stored in a record is the list of nodes given by EQUATION (20):

$$E_{n-1} = \{E_{n-1}^k\}_{1 \leq k \leq K_{n-1}}$$

For each one of these nodes E_{n-1}^k (whose physical equivalent is X_{n-1}^k), its seven children nodes are found in the same fashion as when building the lattice (in particular, node Y_{n-1}^k occupying the center of the descendent cell is found).

The children will have to figure among the $\{E_n^k\}_{1 \leq k \leq K_n}$ as this is what building the lattice during process 900 is meant to insure.

[0096] Next, the values $(C_n^i)_{1 \leq i \leq 7}$ of the derivative instrument that have been previously stored in these seven nodes are retrieved. The drift μ_{n-1}^k is cut by the displacement between X_{n-1}^k and Y_{n-1}^k as shown in EQUATION (28):

$$(28) \quad \mu_{n-1}^k = \mu_{n-1}^k - (Y_{n-1}^k - X_{n-1}^k)$$

All of the above steps are performed in the physical plane.

[0097] Next, the residual drift on the transformed basis is decomposed to obtain

$\begin{pmatrix} \alpha \\ \beta \end{pmatrix}$ by using EQUATION (29):

$$(29) \quad \begin{pmatrix} \alpha \\ \beta \end{pmatrix} = A^{-1} \mu_{n-1}^k$$

The value of the derivative instrument which is to be stored in time slice $n-1$ at node E_{n-1}^k is first evaluated by EQUATION (30):

$$(30) \quad C_{n-1}^k = e^{-r_{n-1}^k \Delta t} \sum_{1 \leq i \leq 7} p_i C_n^i$$

where r_{n-1}^k is the instantaneous interest rate prevailing at node X_{n-1}^k in time slice $n-1$ (if the short rate is among the underlying stochastic variables, r_{n-1}^k will be one of the components of X_{n-1}^k , otherwise it is uniform within the same time slice and equal to the forward rate as given by the initial input of the forward curve).

$(p_i)_{1 \leq i \leq 7}$ are the guaranteed positive risk-neutral probabilities (i.e. the

probabilities that ensure that first and second moment equations of the risk-neutral stochastic processes given above are matched at each node). If the seven children nodes are listed in the same order as the coordinates of the hexagon above, the probabilities are given by EQUATION (31)(a) - (31)(g):

$$\begin{aligned}
 (31)(a) \quad p_1 &= \frac{p}{6} + \frac{\alpha^2}{2} - \frac{\beta^2}{6} + \frac{\alpha}{3} \\
 (31)(b) \quad p_2 &= \frac{p}{6} + \frac{1}{3}\beta^2 + \frac{1}{\sqrt{3}}\alpha\beta + \frac{\alpha}{6} + \frac{\beta}{2\sqrt{3}} \\
 (31)(c) \quad p_3 &= \frac{p}{6} + \frac{1}{3}\beta^2 - \frac{1}{\sqrt{3}}\alpha\beta - \frac{\alpha}{6} + \frac{\beta}{2\sqrt{3}} \\
 (31)(d) \quad p_4 &= \frac{p}{6} + \frac{\alpha^2}{2} - \frac{\beta^2}{6} - \frac{\alpha}{3} \\
 (31)(e) \quad p_5 &= \frac{p}{6} + \frac{1}{3}\beta^2 + \frac{1}{\sqrt{3}}\alpha\beta - \frac{\alpha}{6} - \frac{\beta}{2\sqrt{3}} \\
 (31)(f) \quad p_6 &= \frac{p}{6} + \frac{1}{3}\beta^2 - \frac{1}{\sqrt{3}}\alpha\beta + \frac{\alpha}{6} - \frac{\beta}{2\sqrt{3}}
 \end{aligned}$$

and the probability associated with the center of the descendent cell is given by **EQUATION (31)(g):**

$$(31)(g) \quad p_7 = (1 - p) - (\alpha^2 + \beta^2);$$

and the probability parameter, p, must observe the following constraint:

$$\frac{1}{2} \leq p \leq \frac{2}{3}$$

[0098] Finally, if the derivative instrument is American, C_{n-1}^k is compared to

$I(X_{n-1}^k)$ and the greater of the two is stored.

[0099] This backward recursion will eventually lead to the root X_0 and the time 0, present state, value $C(X_0, 0)$ of the derivative instrument, which is the price of the claim which is contingent on the underlying vector X_0 sought after.

[0100] It is worthy to note that if the procedure is left as described above, the seven children of all the nodes in the lattice will be determined twice--once on the way up while building the lattice and recording the nodes of the different time slices, and once on the way back while computing and storing values of the instrument. Thus, in an embodiment of the present invention, an economy can be

realized on the way up if the forward build up of the lattice is limited to the boundaries of the successive time slices. In other words, in order to determine and record the nodes of the successive time slices, it is sufficient to determine and record the boundary nodes. (In general, children of boundary nodes will lie on the boundaries of the following time slice).

[0101] In sum, the philosophy behind the present invention is to determine, for each node in a time slice, the seven children nodes in the following time slice, such that the first and second moment equations are matched, and the probabilities are positive.

[0102] The lattice is first deformed (by the transformation matrix A) to match the first and second moment equations (EQUATIONS (17) and (18), respectively), with zero drift and the ground probability distribution. In this case, each node gets the hexagonal cell surrounding it (i.e., the natural descendant cell as children. Then the probabilities are modified to match the moment equations with non-zero drift. The greater the drift, the greater the modification of the probabilities.

[0103] However, these probabilities may turn negative. If the drift remains inside the "nucleus," then the probabilities are positive with the children cell being the natural descendant cell.

[0104] If the drift is to be greater, then one has to select another descendant cell. This is designated as the cell inside whose nucleus this "abusive" drift lands. The creation process is prepared to modify the probabilities assigned to the natural cell in order to match this drift, but those probabilities were going to be negative. By having an alternative cell, the probabilities can be computed otherwise. This is accomplished by utilizing the alternative cell as though it were a natural descendant cell. That is, as far as the probabilities are concerned, the process proceeds as if the father node were transported to the center of the alternative cell. In other words, the process proceeds as if the drift was the vector joining the center of the alternative cell to the point of impact of the old drift. This is termed "cutting the drift." The probabilities are then computed. At the end of the process, however, the "real" father node is still where it was--its children are vertices of the

alternative cell (which is farther away than its natural descendant), and the probabilities are as computed.

VII. Details of Lattice Geometry - Three Dimensional Case

[0105] Having generally described the lattice creation process 900 and lattice traversal process 1000 above, a more detailed description is now given for the case where pricing a basket option which includes three underlying assets is desired.

A. Inputs

[0106] First, three underlying stochastic variables representing each asset in the basket are considered: x_1 , x_2 and x_3 . This comprises one underlying stochastic vector X :

$$X = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}$$

The stochastic processes in the risk-neutral world are then given by:

$$\begin{aligned} dx_1 &= \mu_1(X, t)dt + \sigma_1 dz_1 \\ dx_2 &= \mu_2(X, t)dt + \sigma_2 dz_2 \\ dx_3 &= \mu_3(X, t)dt + \sigma_3 dz_3 \end{aligned}$$

With correlation coefficients:

ρ_{12} between dz_1 and dz_2

ρ_{13} between dz_1 and dz_3

ρ_{23} between dz_2 and dz_3

The volatility parameters are:

$$\sigma_1 \sigma_2 \sigma_3$$

and the correlation coefficient are constant, but the drift vector is given by:

$$\mu = \begin{pmatrix} \mu_1 \\ \mu_2 \\ \mu_3 \end{pmatrix}$$

The drift Vector μ depends on both the state X and time t . The initial state at time zero (0) is:

$$X_0 = \begin{pmatrix} x_1^0 \\ x_2^0 \\ x_3^0 \end{pmatrix}$$

[0107] The present invention aims to price, at time $t = 0$, a contingent claim C (i.e. a derivative instrument whose payoff is contingent upon the underlying vector X), of time to maturity T and payoff at maturity $P(X)$. The price at time $t=0$, is denoted by $C(X_0, 0)$. If the derivative instrument is an American-type option, its time t , state X , price $C(X, t)$ must verify:

$$C(X, t) \geq I(X)$$

[0108] Therefore, the inputs of the lattice building process are: (a) the diffusion parameters:

$$\rho_{ij} \sigma_i$$

(b) The complete drift function:

$$\mu(X, t) = \begin{pmatrix} \mu_1(X, t) \\ \mu_2(X, t) \\ \mu_3(X, t) \end{pmatrix}$$

(c) the initial value X_0 of the underlying state vector; (d) the value of the short interest rate r (or the interest rate term structure - yield curve or forward curve); (e) the number of time steps N one wishes to take; and (f) the probability parameters: p_1^0 and p_2^0 .

[0109] Once N is fixed, the elementary time step is:

$$\Delta t = \frac{T}{N}$$

B. Determination of the Geometry of the Elementary Cell

[0110] The elementary cell is a centered rhombic dodecahedron. However, it must be geometrically deformed in order to reflect the diffusion matrix:

$$\Sigma = \begin{pmatrix} \sigma_1^2 & \rho_{12}\sigma_1\sigma_2 & \rho_{13}\sigma_1\sigma_3 \\ \rho_{12}\sigma_1\sigma_2 & \sigma_2^2 & \rho_{23}\sigma_2\sigma_3 \\ \rho_{13}\sigma_1\sigma_3 & \rho_{23}\sigma_2\sigma_3 & \sigma_3^2 \end{pmatrix} \Delta t$$

The transformation matrix is:

$$A = \frac{1}{\sqrt{\frac{4}{3} p_1^0 + p_2^0}} P \sqrt{\Lambda}$$

where

$$\Sigma = P \Lambda P^*$$

That is:

$$\Lambda = \begin{pmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{pmatrix}$$

is the matrix of eigenvalues of S , and P the matrix of eigenvectors, which are computed by standard techniques as will be apparent to one skilled in the relevant art(s). The “*” is for the transpose of a matrix.

[0111] The lattice, that is the set of states (or nodes):

$$X = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}$$

is a deformed rhombic dodecahedral lattice. It is the aim to retain it for the discretization. However, we define a straight computational lattice E where coordinates of the nodes have integer values and correspond one to one to the real, "physical" nodes:

$$E = \begin{pmatrix} i_1 \\ i_2 \\ i_3 \end{pmatrix} \leftrightarrow X = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}$$

through the transformation:

$$X = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = A \begin{pmatrix} i_1 \\ i_2 \\ i_3 \end{pmatrix}$$

Indeed, in the computational plane, the rhombic dodecahedron is represented as shown in **FIG. 15**, with its vertices having the following integer coordinates:

$$H_0^{dodeca} = \begin{pmatrix} 2 & 0 & 0 & 1 & 1 & -1 & 1 & 0 & -2 & 0 & 0 & -1 & -1 & 1 & -1 & 0 \\ 0 & 2 & 0 & 1 & -1 & 1 & 1 & 0 & 0 & -2 & 0 & -1 & 1 & -1 & -1 & 0 \\ 0 & 0 & 2 & 1 & 1 & 1 & -1 & 0 & 0 & 0 & -2 & -1 & -1 & -1 & 1 & 0 \end{pmatrix}$$

and being numbered:

$$(1 \ 2 \ 3 \ 4 \ 5 \ 6 \ 7 \ 8 \ 1' \ 2' \ 3' \ 4' \ 5' \ 6' \ 7' \ 8')$$

One will note that, in an embodiment of the present invention, the center node has been counted twice for symmetry reasons.

C. Building of the Lattice

[0112] The building of the lattice is a recursive procedure running forward in time. It starts with the origin (or root):

$$X_0 = \begin{pmatrix} x_1^0 \\ x_2^0 \\ x_3^0 \end{pmatrix}$$

and progressively "marks", for each time slice n ($0 \leq n \leq N$), the nodes of the computational lattice E which will belong to this time slice. It is the drift function that will of course govern this process.

[0113] The recursive mechanism is the following:

[0114] If nodes of time slice $n - 1$ are:

$$E_{n-1} = \{E_{n-1}^k\}_{1 \leq k \leq K_{n-1}}$$

where K_{n-1} is their number, the following steps (a)-(d) are performed:

(a) for each node E_{n-1}^k , find its physical correspondent X_{n-1}^k by computing:

$$X_{n-1}^k = X_0 + AE_{n-1}^k$$

(This means the root X_0 has the computational correspondent $E_0 = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$.)

(b) Compute the drift vector at X_{n-1}^k by computing:

$$\mu(X_{n-1}^k, (n-1)\Delta t) = \begin{pmatrix} \mu_1(X_{n-1}^k, (n-1)\Delta t) \\ \mu_2(X_{n-1}^k, (n-1)\Delta t) \\ \mu_3(X_{n-1}^k, (n-1)\Delta t) \end{pmatrix} \Delta t$$

(c) The drift vector sends us to a point Z_{n-1}^k somewhere in the physical plane:

$$Z_{n-1}^k = X_{n-1}^k + \mu_{n-1}^k$$

and the aim becomes to determine the node of the lattice that is "closest" (as explained herein) to Z_{n-1}^k . This is called the "target" node.

[0115] Y_{n-1}^k is the candidate node for occupying the center of the cell of descendants of X_{n-1}^k . Meaning, once we determine Y_{n-1}^k , the fourteen other children nodes of X_{n-1}^k will be determined as well, as vertices of the rhombic dodecahedron surrounding Y_{n-1}^k .

[0116] The determining criterion is that the point where the drift sends us, Z_{n-1}^k , lie in the nucleus of the target cell. The nucleus of the rhombic dodecahedron is the solid called Lord Kelvin's solid and shown in FIG. 16. That is, Y_{n-1}^k must be the center of the rhombic dodecahedron in whose nucleus Z_{n-1}^k lies.

Calling $\begin{pmatrix} z_1 \\ z_2 \\ z_3 \end{pmatrix}$ the physical coordinates of Z_{n-1}^k , its coordinates in the computational plane are:

$$e = \begin{pmatrix} e_1 \\ e_2 \\ e_3 \end{pmatrix} = A^{-1} \begin{pmatrix} z_1 - x_1^0 \\ z_2 - x_2^0 \\ z_3 - x_3^0 \end{pmatrix}$$

These will not generally be integer values, so Y_{n-1}^k will be the one among the nodes of the computational space which is closest to Z_{n-1}^k . "Closest" here means "closest in terms of Euclidian distance" as measured *in computational space*.

[0117] The selected minimizer f will then represent the coordinates in the computational plane of the target node Y_{n-1}^k . As for the fourteen other children, their coordinates in the computational plane will be given by:

$$f + \begin{pmatrix} 2 & 0 & 0 & 1 & 1 & -1 & 1 & -2 & 0 & 0 & -1 & -1 & 1 & -1 \\ 0 & 2 & 0 & 1 & -1 & 1 & 1 & 0 & -2 & 0 & -1 & 1 & -1 & -1 \\ 0 & 0 & 2 & 1 & 1 & 1 & -1 & 0 & 0 & -2 & -1 & -1 & -1 & 1 \end{pmatrix}$$

(d) Repeating this procedure for each one of E_{n-1}^k , and discarding repetitions, will give us the list of the nodes of the computational lattice which will constitute time slice n :

$$E_n = \{E_n^k\}_{1 \leq k \leq K_n} = \bigcup_{1 \leq k \leq K_{n-1}} \{\text{fifteen children of } E_{n-1}^k\}$$

D. Traversing the Lattice

[0118] When building of the lattice is completed, a record for each time slice of the nodes constituting the lattice is obtained. In particular, the complete list of nodes of the last time slice N is now available:

$$E_N = \{E_N^k\}_{1 \leq k \leq K_N}$$

The payoff of the derivative instrument is then computed at this last time slice. For each one of the nodes, find its physical coordinates through:

$$X_N^k = X_0 + AE_N^k$$

and compute the corresponding payoff $P(X_N^k)$. This is the value of the derivative instrument which will be stored in time slice N at node E_N^k .

[0119] Rolling back in the tree will then inductively takes place as follows. First, assuming values of the derivative instrument have been computed and stored at all

nodes E_n^k of time slice n , step back to time slice $n - 1$. In the list of nodes, the following is already stored:

$$E_{n-1} = \left\{ E_{n-1}^k \right\}_{1 \leq k \leq K_{n-1}}$$

Thus, for each one of the nodes E_{n-1}^k in the list of nodes (whose physical equivalent is X_{n-1}^k), find its fifteen children nodes in the same fashion as while building the lattice (in particular find node Y_{n-1}^k occupying the center of the descendent cell). The children will have to figure among the $\left\{ E_n^k \right\}_{1 \leq k \leq K_n}$ (this is what building the lattice is meant to insure).

[0120] Retrieve the values $\left(C_n^i \right)_{1 \leq i \leq 7}$ of the derivative instrument that have been previously stored in these seven nodes. Next, the drift μ_{n-1}^k is cut by the displacement between X_{n-1}^k and Y_{n-1}^k :

$$\mu_{n-1}^k = \mu_{n-1}^k - \left(Y_{n-1}^k - X_{n-1}^k \right)$$

(all this is done in the physical space).

Then, decompose the residual drift on the transformed basis to get $\begin{pmatrix} \alpha \\ \beta \\ \gamma \end{pmatrix}$, where:

$$\begin{pmatrix} \alpha \\ \beta \\ \gamma \end{pmatrix} = A^{-1} \mu_{n-1}^k$$

The value of the derivative instrument which is to be stored in time slice $n - 1$ at node E_{n-1}^k is first evaluated as:

$$C_{n-1}^k = e^{-r_{n-1}^k \Delta t} \sum_{1 \leq i \leq 15} p_i C_n^i$$

where r_{n-1}^k is the instantaneous interest rate prevailing at node X_{n-1}^k in time slice $n-1$. (If the short rate is among the underlying stochastic variables, r_{n-1}^k will be one of the components of X_{n-1}^k , otherwise it is uniform within the same time slice and equal to the forward rate as given by the initial input of the forward curve.)

[0121] $(p_i)_{1 \leq i \leq 15}$ are the guaranteed positive, risk-neutral probabilities (i.e., the probabilities that ensure that first and second moment equations of the risk-neutral stochastic processes given above are matched at each node). If the fifteen children nodes are listed in the same order as the coordinates of the rhombic dodecahedron above, computing the probabilities proceeds in two stages.

[0122] It is first assumed that (α, β, γ) has all three positive components and that $\gamma \geq \alpha \geq \beta$. The sums of probabilities assigned to symmetrical nodes are then expressed as follows (i.e., $\pi_i = p_i + p_1$):

$$\pi_1 = \frac{p_1^0}{3} + \lambda \alpha^2 + \theta(\beta^2 + \gamma^2)$$

$$\pi_2 = \frac{p_1^0}{3} + \lambda \beta^2 + \theta(\alpha^2 + \gamma^2)$$

$$\pi_3 = \frac{p_1^0}{3} + \lambda \gamma^2 + \theta(\alpha^2 + \beta^2)$$

$$\pi_4 = \frac{p_2^0}{4} + \eta(\alpha^2 + \beta^2 + \gamma^2) + \varphi(\alpha\beta + \alpha\gamma + \beta\gamma)$$

$$\pi_5 = \frac{p_2^0}{4} + \eta(\alpha^2 + \beta^2 + \gamma^2) + \varphi(-\alpha\beta + \alpha\gamma - \beta\gamma)$$

$$\pi_6 = \frac{p_2^0}{4} + \eta(\alpha^2 + \beta^2 + \gamma^2) + \varphi(-\alpha\beta - \alpha\gamma + \beta\gamma)$$

$$\pi_7 = \frac{p_2^0}{4} + \eta(\alpha^2 + \beta^2 + \gamma^2) + \varphi(\alpha\beta - \alpha\gamma - \beta\gamma)$$

$$\pi_8 = p_3^0 + \rho(\alpha^2 + \beta^2 + \gamma^2)$$

(π_8 is the probability attached to the center.); where:

$$\lambda = -\eta + \frac{1}{4}$$

$$\theta = -\eta$$

$$\varphi = \frac{1}{4}$$

$$\rho = -\eta - \frac{1}{4}$$

and observing the following constraints:

$$\eta \leq \frac{11}{80} - \frac{p_2^0}{5}$$

$$0 \leq p_2^0 \leq \frac{1}{16}$$

$$\frac{1}{8} \leq \eta \leq \frac{11}{80} - \frac{p_2^0}{5}$$

$$\frac{15}{32} \leq p_1^0 \leq \frac{17}{32} - p_2^0$$

$$\frac{15}{32} \leq p_3^0 \leq \frac{17}{32} - p_2^0$$

$$p_3^0 = 1 - p_1^0 - p_2^0 ; \text{ and}$$

the individual probabilities will then be expressed by:

$$p_i = \frac{\pi_i}{2} + \omega_i \text{ and } p_i' = \frac{\pi_i}{2} - \omega_i \text{ where:}$$

$$\omega_1 = a\alpha$$

$$\omega_2 = a\beta$$

$$\omega_3 = a\gamma$$

$$\omega_4 = b(\alpha + \beta + \gamma)$$

$$\omega_5 = b(\alpha - \beta + \gamma)$$

$$\omega_6 = b(-\alpha + \beta + \gamma)$$

$$\omega_7 = b(\alpha + \beta - \gamma)$$

and

$$2a + 4b = \frac{1}{2}$$

[0123] The strategy of the present invention then consists in fixing the remaining parameters, for instance:

$$\eta = \frac{1}{8}$$

$$p_1^0 = \frac{15}{4} \eta = \frac{15}{32}$$

$$p_2^0 = \frac{1}{16}$$

$$p_3^0 = \frac{15}{32}$$

$$a = 0, \text{ so that } b = 1/8$$

The above gives a temporary expression of the individual probabilities. If they are all positive, then computation of the probabilities at the node is complete. Otherwise, to finalize the expression of probabilities (i.e., in order to make sure the probabilities are positive), each probability must be shifted by an amount dp_j , which is computed as follows:

- if, for $j \in \{4,5,6\}$, one of the \tilde{p}_j' is negative, set:

$$dp_j' = \tilde{p}_j'$$

(i.e., bring the negative probability back to zero)

$$dp_j = \tilde{p}_j'$$

(i.e., maintain constant the sums of conjugate probabilities)

Otherwise, $dp_j' = dp_j = 0$

Then, set:

$$dp_7 = -\tilde{p}_7$$

$$dp_7' = \tilde{p}_7$$

and finally, calculate:

$$\begin{aligned} dp_1 &= \frac{-dp_4 - dp_5 + dp_6 - dp_7}{2} \\ dp_2 &= \frac{-dp_4 + dp_5 - dp_6 - dp_7}{2} \\ dp_3 &= \frac{-dp_4 - dp_5 - dp_6 + dp_7}{2} \end{aligned}$$

and

$(dp_j = -dp_j)_{j \in \{1,2,3\}}$. The probability at the center node is left untouched.

In the event that our first assumption is not true (i.e., all three components

of $\begin{pmatrix} \alpha \\ \beta \\ \gamma \end{pmatrix}$ are not positive and that the expression $\gamma \geq \alpha \geq \beta$ is not true), it is recalled

that the vertices of the rhombic dodecahedron (with the center node counted twice) and they are now numbered in the following fashion:

$$H_0^{dodeca} = \begin{pmatrix} 2 & 0 & 0 & 1 & 1 & -1 & 1 & 0 & -2 & 0 & 0 & -1 & -1 & 1 & -1 & 0 \\ 0 & 2 & 0 & 1 & -1 & 1 & 1 & 0 & 0 & -2 & 0 & -1 & 1 & -1 & -1 & 0 \\ 0 & 0 & 2 & 1 & 1 & 1 & -1 & 0 & 0 & 0 & -2 & -1 & -1 & -1 & 1 & 0 \end{pmatrix}$$

And we now number them in the following fashion:

(1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16)

Now, **prob** is defined to be the array of probabilities as assigned to the vertices in this order (splitting in two the probability for the two twin-center nodes). Then the following eight transformations are defined (using pseudo-code) as follows:

```
salpha(prob);  
let probtemp=prob;  
then reshuffle thus:  
prob[1]=probtemp[9];  
prob[2]=probtemp[2];  
prob[3]=probtemp[3];  
prob[4]=probtemp[6];  
prob[5]=probtemp[15];  
prob[6]=probtemp[4];  
prob[7]=probtemp[13];  
prob[9]=probtemp[1];  
prob[10]=probtemp[10];  
prob[11]=probtemp[11];  
prob[12]=probtemp[14];  
prob[13]=probtemp[7];  
prob[14]=probtemp[12];  
prob[15]=probtemp[5];  
return(prob);
```

```
sgam(prob);  
let probtemp=prob;  
then reshuffle thus:  
prob[1]=probtemp[1];  
prob[2]=probtemp[2];  
prob[3]=probtemp[11];  
prob[4]=probtemp[7];  
prob[5]=probtemp[14];  
prob[6]=probtemp[13];  
prob[7]=probtemp[4];  
prob[9]=probtemp[9];  
prob[10]=probtemp[10];  
prob[11]=probtemp[3];  
prob[12]=probtemp[15];  
prob[13]=probtemp[6];  
prob[14]=probtemp[5];  
prob[15]=probtemp[12];  
return(prob);
```

```
sbeta(prob);  
let probtemp=prob;  
then reshuffle thus:  
prob[1]=probtemp[1];  
prob[2]=probtemp[10];  
prob[3]=probtemp[3];  
prob[4]=probtemp[5];  
prob[5]=probtemp[4];  
prob[6]=probtemp[15];  
prob[7]=probtemp[14];  
prob[9]=probtemp[9];  
prob[10]=probtemp[2];  
prob[11]=probtemp[11];  
prob[12]=probtemp[13];  
prob[13]=probtemp[12];  
prob[14]=probtemp[7];  
prob[15]=probtemp[6];  
return(prob);
```

```
salbeta(prob);  
let probtemp=prob;  
then reshuffle thus:  
prob[1]=probtemp[2];  
prob[2]=probtemp[1];  
prob[3]=probtemp[3];  
prob[4]=probtemp[4];  
prob[5]=probtemp[6];  
prob[6]=probtemp[5];  
prob[7]=probtemp[7];  
prob[9]=probtemp[10];  
prob[10]=probtemp[9];  
prob[11]=probtemp[11];  
prob[12]=probtemp[12];  
prob[13]=probtemp[14];  
prob[14]=probtemp[13];  
prob[15]=probtemp[15];  
return(prob);
```

```
sbetagam(prob);  
let probtemp=prob;  
then reshuffle thus:  
prob[1]=probtemp[1];  
prob[2]=probtemp[3];  
prob[3]=probtemp[2];  
prob[4]=probtemp[4];  
prob[5]=probtemp[7];  
prob[6]=probtemp[6];  
prob[7]=probtemp[5];  
prob[9]=probtemp[9];  
prob[10]=probtemp[11];  
prob[11]=probtemp[10];  
prob[12]=probtemp[12];  
prob[13]=probtemp[15];  
prob[14]=probtemp[14];  
prob[15]=probtemp[13];  
return(prob);
```

```
rotpos(prob);  
let probtemp=prob;  
then reshuffle thus:  
prob[1]=probtemp[2];  
prob[2]=probtemp[3];  
prob[3]=probtemp[1];  
prob[4]=probtemp[4];  
prob[5]=probtemp[7];  
prob[6]=probtemp[5];  
prob[7]=probtemp[6];  
prob[9]=probtemp[10];  
prob[10]=probtemp[11];  
prob[11]=probtemp[9];  
prob[12]=probtemp[12];  
prob[13]=probtemp[15];  
prob[14]=probtemp[13];  
prob[15]=probtemp[14];  
return(prob);
```

```
salgam(prob);  
let probtemp=prob;  
then reshuffle thus:  
prob[1]=probtemp[3];  
prob[2]=probtemp[2];  
prob[3]=probtemp[1];  
prob[4]=probtemp[4];  
prob[5]=probtemp[5];  
prob[6]=probtemp[7];  
prob[7]=probtemp[6];  
prob[9]=probtemp[11];  
prob[10]=probtemp[10];  
prob[11]=probtemp[9];  
prob[12]=probtemp[12];  
prob[13]=probtemp[13];  
prob[14]=probtemp[15];  
prob[15]=probtemp[14];  
return(prob);
```

```
rotneg(prob);  
let probtemp=prob;  
then reshuffle thus:  
prob[1]=probtemp[3];  
prob[2]=probtemp[1];  
prob[3]=probtemp[2];  
prob[4]=probtemp[4];  
prob[5]=probtemp[6];  
prob[6]=probtemp[7];  
prob[7]=probtemp[5];  
prob[9]=probtemp[11];  
prob[10]=probtemp[9];  
prob[11]=probtemp[10];  
prob[12]=probtemp[12];  
prob[13]=probtemp[14];  
prob[14]=probtemp[15];  
prob[15]=probtemp[13];  
return(prob);
```

The above eight transformations allow probabilities within the same array to be

permuted. Now, any one of the components of $\begin{pmatrix} \alpha \\ \beta \\ \gamma \end{pmatrix}$ can be negative, and thus

must be so marked. Thus, a marker-variable is initialized, **case = 0**, and the following logic is performed:

- if $\alpha < 0$, set $\alpha = -\alpha$ and $\text{case}=\text{case}+1$;
- if $\beta < 0$, set $\beta = -\beta$ and $\text{case}=\text{case}+10$;
- if $\gamma < 0$, set $\gamma = -\gamma$ and $\text{case}=\text{case}+100$;

[0124] This allows a fall back to the case of three positive components, and the marker-variable indicates how this was achieved. For instance, if case exits with the value 101: this indicates that the sign of alpha and gamma were changed. Likewise, a value of 111 indicates the sign of alpha, beta and gamma were changed, and so forth.

[0125] Therefore, the above results in three positive components which fall into one of the following six cases, which are processed as follows:

1. Case 1

[0126] If $(\alpha \geq \beta \geq \gamma)$, then permute the three components in order to fall back to the case where $\gamma \geq \alpha \geq \beta$. In other words, set $\gamma = \alpha$; $\alpha = \beta$; and $\beta = \gamma$:

[0127] The process for computing the guaranteed positive individual probabilities, as given above, can then be applied. However, reassignment of the probabilities to a permutation of the nodes of the rhombic dodecahedron is required. This is done as follows:

```
if case==0;
prob=rotneg(prob);
elseif case==1;
prob=salpha(rotneg(prob));
elseif case==10;
prob=sbeta(rotneg(prob));
elseif case==100;
prob=sgam(rotneg(prob));
elseif case==11;
prob=salpha(sbeta(rotneg(prob)));
elseif case==101;
prob=salpha(sgam(rotneg(prob)));
elseif case==110;
prob=sbeta(sgam(rotneg(prob)));
elseif case==111;
prob=salpha(sbeta(sgam(rotneg(prob))));
endif.
```

2. Case 2

[0128] If ($\alpha \geq \gamma \geq \beta$), then set $\gamma = \alpha$; and $\alpha = \gamma$. The array of probabilities is computed and then:

```
if case==0;
prob=salgam(prob);
elseif case==1;
prob=salpha(salgam(prob));
elseif case==10;
prob=sbeta(salgam(prob));
elseif case==100;
prob=sgam(salgam(prob));
elseif case==11;
prob=salpha(sbeta(salgam(prob)));
elseif case==101;
prob=salpha(sgam(salgam(prob)));
elseif case==110;
prob=sbeta(sgam(salgam(prob)));
elseif case==111;
prob=salpha(sbeta(sgam(salgam(prob))));
endif.
```

3. Case 3

[0129] If ($\beta \geq \alpha \geq \gamma$), then set $\gamma = \beta$; and $\beta = \gamma$: The array of probabilities is computed, and then:

```
if case==0;
prob=sbetagam(prob);
elseif case==1;
prob=salpha(sbetagam(prob));
elseif case==10;
prob=sbeta(sbetagam(prob));
elseif case==100;
prob=sgam(sbetagam(prob));
elseif case==11;
prob=salpha(sbeta(sbetagam(prob)));
elseif case==101;
prob=salpha(sgam(sbetagam(prob)));
elseif case==110;
prob=sbeta(sgam(sbetagam(prob)));
elseif case==111;
prob=salpha(sbeta(sgam(sbetagam(prob))));
endif.
```

4. Case 4

[0130] If ($\beta \geq \gamma \geq \alpha$), then set $\gamma = \beta$; $\alpha = \gamma$; and $\beta = \alpha$. The array of probabilities is computed, and then:

```
if case==0;
prob=rotpos(prob);
elseif case==1;
prob=salpha(rotpos(prob));
elseif case==10;
prob=sbeta(rotpos(prob));
elseif case==100;
prob=sgam(rotpos(prob));
elseif case==11;
prob=salpha(sbeta(rotpos(prob)));
elseif case==101;
prob=salpha(sgam(rotpos(prob)));
elseif case==110;
prob=sbeta(sgam(rotpos(prob)));
elseif case==111;
prob=salpha(sbeta(sgam(rotpos(prob))));
endif.
```

5. Case 5

[0131] If ($\gamma \geq \alpha \geq \beta$), then compute the array of probabilities, and then:

```
if case==1;
prob=salpha(prob);
elseif case==10;
prob=sbeta(prob);
elseif case==100;
prob=sgam(prob);
elseif case==11;
prob=salpha(sbeta(prob));
elseif case==101;
prob=salpha(sgam(prob));
elseif case==110;
prob=sbeta(sgam(prob));
elseif case==111;
prob=salpha(sbeta(sgam(prob)));
endif.
```

6. Case 6

[0132] If, $(\gamma \geq \beta \geq \alpha)$, then set $\alpha = \beta$; and $\beta = \alpha$. The array of probabilities is computed, and then:

```

if case==0;
prob=salbeta(prob);
elseif case==1;
prob=salpha(salbeta(prob));
elseif case==10;
prob=sbeta(salbeta(prob));
elseif case==100;
prob=sgam(salbeta(prob));
elseif case==11;
prob=salpha(sbeta(salbeta(prob)));
elseif case==101;
prob=salpha(sgam(salbeta(prob)));
elseif case==110;
prob=sbeta(sgam(salbeta(prob)));
elseif case==111;
prob=salpha(sbeta(sgam(salbeta(prob))));
endif.

```

This completes the computation of the individual probabilities assigned to the

rhombic dodecahedral vertices in all cases where $\begin{pmatrix} \alpha \\ \beta \\ \gamma \end{pmatrix}$ lies in the nucleus.

[0133] Finally, if the derivative instrument is an American-type, C_{n-1}^k is compared to $I(X_{n-1}^k)$ and we store the greater of the two. This backward recursion will eventually lead us to the root X_0 and the time 0, present state, value $C(X_0, 0)$ of the derivative instrument, which is the theoretical price sought by the present invention.

[0134] If the embodiment described above is left as is, the fifteen children of all the nodes in the lattice will be determined twice over--once on our way up while building the lattice and recording the nodes of the different time slices, and once

on the way back while computing and storing values of the instrument. An economy can be realized, in an embodiment of the present invention, on the way up. If the forward build up of the lattice is limited to the boundaries of the successive time slices. In other words, in order to determine and record the nodes of the successive time slices, it is sufficient to determine and records the boundary nodes. (In general, children of boundary nodes will lie on the boundaries of the following time slice.)

VIII. Computational Efficiency of the Present Invention

[0135] As mentioned above with reference to TABLE 1, the present invention extends Hull and White's trinomial technique to obtain the most economical and efficient lattice discretization for higher dimensional Brownian motions of the most general form (i.e., EQUATION (14)). More specifically, consider the two-and three-dimensional cases.

[0136] In the two-dimension case (i.e., a basket option with two assets underlying it), the hexagonal lattice is the optimal two-dimensional lattice because there is no way of further reducing the number of nodes in the elementary cell, and the number of nodes of the T^{th} slice will here be:

$$(3T^2 + 3T + 1)$$

as opposed to:

$$(4T^2 + 4T + 1)$$

in the Hull and White tensor product lattice. This gain is significant, but not as significant as in the three-dimensional case described below. Nevertheless, computing time reduced not only because the backward recursion has to read much less nodes in each time slice where the prices of the option are computed, but also because the computation of each single price - as discounted expected value of prices in descendant nodes - will require less calls of previously computed prices (7 instead of 9 as highlighted in TABLE 1).

[0137] In the three-dimensional case (i.e., a basket option with three assets underlying it), the number of nodes in the T^{th} time slice is:

$$(2T + 1)(2T^2 + 2T + 1)$$

For example, at the second time slice, 65 nodes are present in the rhombic dodecahedral lattice as shown in FIG. 13. Despite its apparent complexity, the lattice is in fact simpler than the traditional grid. More specifically, the number of nodes in the case of the Hull and White trinomial tensor product is:

$$(2T + 1)(4T^2 + 4T + 1)$$

The ratio to the present invention is:

$$\frac{1}{2} + \frac{1}{2(4T^2 + 4T + 1)}$$

and converges rapidly to one-half. Bearing in mind that the number of nodes in a cell (15) is almost half the number of nodes of the tensor product cell (27) (as highlighted in TABLE 1), the gain in computing time is then approximately fourfold.

IX. Example Implementations

[0138] The present invention (e.g., lattice creation process 900, lattice traversal process 1000, or any part thereof) may be implemented using hardware, software or a combination thereof and may be implemented in one or more computer systems or other processing systems. In fact, in one embodiment, the invention is directed toward one or more computer systems capable of carrying out the functionality described herein. An example of a computer system 1400 is shown in FIG. 14. The computer system 1400 includes one or more processors, such as processor 1404. The processor 1404 is connected to a communication infrastructure 1406 (e.g., a communications bus, cross-over bar, or network). Various software embodiments are described in terms of this exemplary computer system. After reading this description, it will become apparent to a person skilled

in the relevant art how to implement the invention using other computer systems and/or computer architectures.

[0139] Computer system 1400 can include a display interface 1405 that forwards graphics, text, and other data from the communication infrastructure 1402 (or from a frame buffer not shown) for display on the display unit 1430.

[0140] Computer system 1400 also includes a main memory 1408, preferably random access memory (RAM), and may also include a secondary memory 1410. The secondary memory 1410 may include, for example, a hard disk drive 1412 and/or a removable storage drive 1414, representing a floppy disk drive, a magnetic tape drive, an optical disk drive, etc. The removable storage drive 1414 reads from and/or writes to a removable storage unit 1418 in a well-known manner. Removable storage unit 1418, represents a floppy disk, magnetic tape, optical disk, etc. which is read by and written to by removable storage drive 1414. As will be appreciated, the removable storage unit 1418 includes a computer usable storage medium having stored therein computer software and/or data.

[0141] In alternative embodiments, secondary memory 1410 may include other similar means for allowing computer programs or other instructions to be loaded into computer system 1400. Such means may include, for example, a removable storage unit 1422 and an interface 1420. Examples of such may include a program cartridge and cartridge interface (such as that found in video game devices), a removable memory chip (such as an EPROM, or PROM) and associated socket, and other removable storage units 1422 and interfaces 1420 which allow software and data to be transferred from the removable storage unit 1422 to computer system 1400.

[0142] Computer system 1400 may also include a communications interface 1424. Communications interface 1424 allows software and data to be transferred between computer system 1400 and external devices. Examples of communications interface 1424 may include a modem, a network interface (such as an Ethernet card), a communications port, a PCMCIA slot and card, etc. Software and data transferred via communications interface 1424 are in the form

of signals 1428 which may be electronic, electromagnetic, optical or other signals capable of being received by communications interface 1424. These signals 1428 are provided to communications interface 1424 via a communications path (i.e., channel) 1426. This channel 1426 carries signals 1428 and may be implemented using wire or cable, fiber optics, a phone line, a cellular phone link, an RF link and other communications channels.

[0143] In this document, the terms “computer program medium” and “computer usable medium” are used to generally refer to media such as removable storage drive 1414, a hard disk installed in hard disk drive 1412, and signals 1428. These computer program products are means for providing software to computer system 1400. The invention is directed to such computer program products.

[0144] Computer programs (also called computer control logic) are stored in main memory 1408 and/or secondary memory 1410. Computer programs may also be received via communications interface 1424. Such computer programs, when executed, enable the computer system 1400 to perform the features of the present invention as discussed herein. In particular, the computer programs, when executed, enable the processor 1404 to perform the features of the present invention. Accordingly, such computer programs represent controllers of the computer system 1400.

[0145] In an embodiment where the invention is implemented using software, the software may be stored in a computer program product and loaded into computer system 1400 using removable storage drive 1414, hard drive 1412 or communications interface 1424. The control logic (software), when executed by the processor 1404, causes the processor 1404 to perform the functions of the invention as described herein.

[0146] In another embodiment, the invention is implemented primarily in hardware using, for example, hardware components such as application specific integrated circuits (ASICs). Implementation of the hardware state machine so as to perform the functions described herein will be apparent to persons skilled in the relevant art(s).

[0147] In yet another embodiment, the invention is implemented using a combination of both hardware and software.

X. Conclusion

[0148] While various embodiments of the present invention have been described above, it should be understood that they have been presented by way of example, and not limitation. It will be apparent to persons skilled in the relevant art(s) that various changes in form and detail can be made therein without departing from the spirit and scope of the invention. Thus the present invention should not be limited by any of the above-described exemplary embodiments, but should be defined only in accordance with the following claims and their equivalents.